# Data Visualization Beginner’s Guide

Data visualization is the graphical representation of data and information using visual elements like charts, graphs, maps, and plots.

# MatpotLib

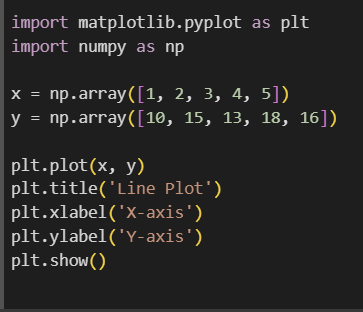
Matplotlib is a foundational Python plotting library offering precise control over every plot aspect. It’s powerful for creating a wide variety of static, animated, and interactive plots. It is one of the most powerful and important tools for data visualization.

# Use Cases of Matpotlib

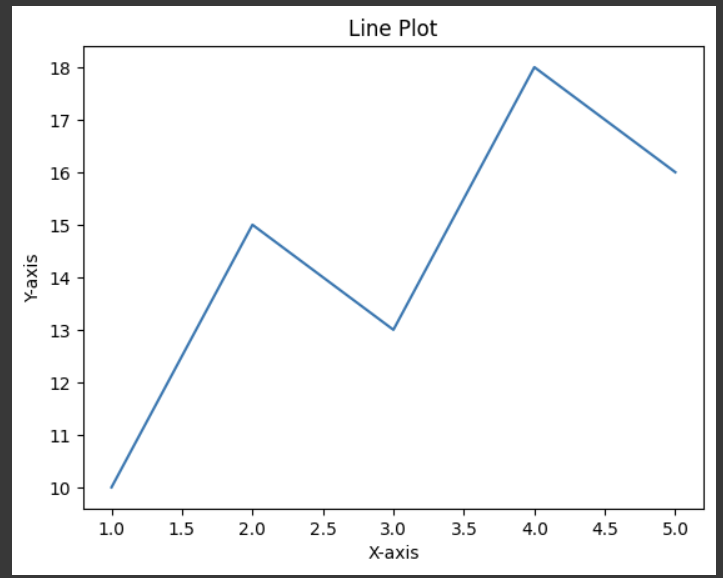
Line Chart

Use Case: To visualize trends over a period.

Code Snippet:



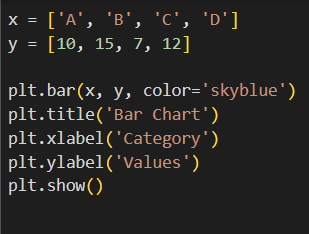
Output:



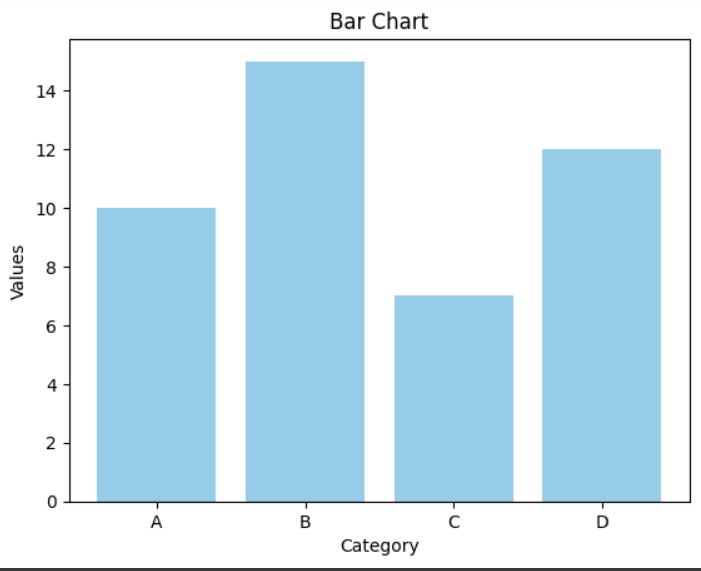
Bar Chart

Use Case: To compare quantities among different categories.

Code Snippet:



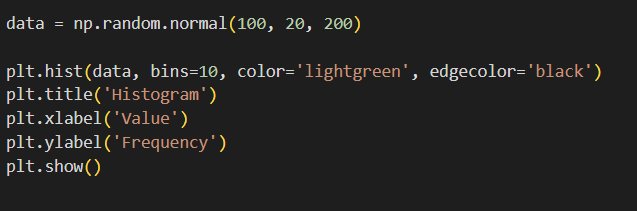
Output:



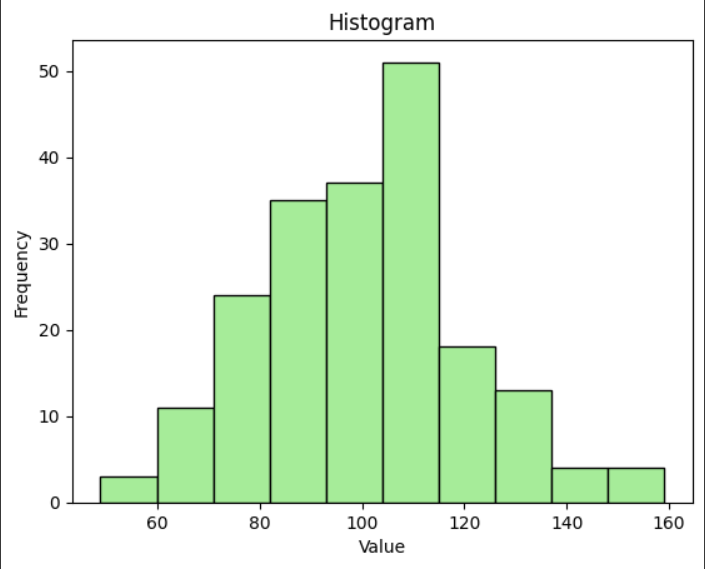
Histogram

Use Case: To observe the frequency distribution of a dataset.

Code Snippet:



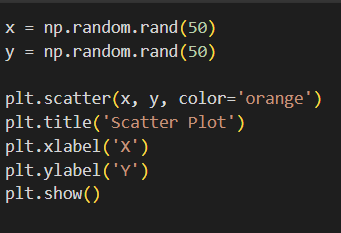
Output:



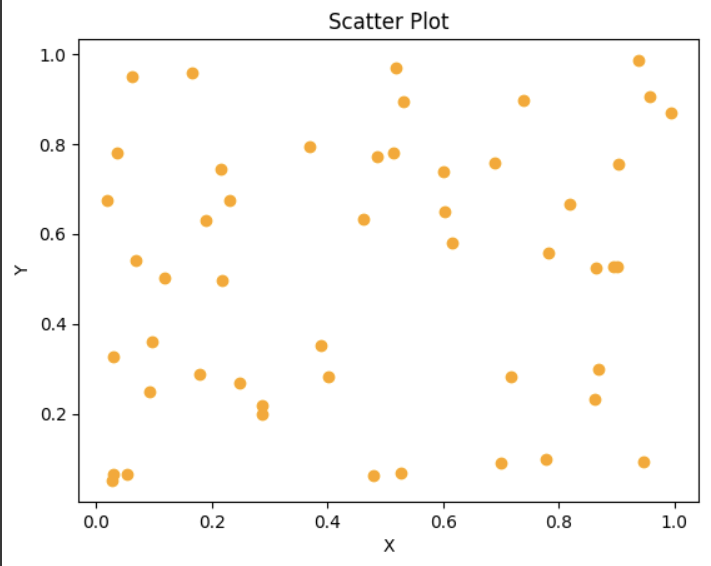
Scatter Plot

Use Case: To show the relationship between two continuous variables.

Code Snippet:



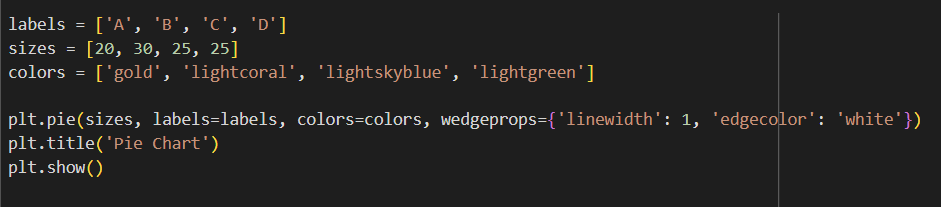
Output:



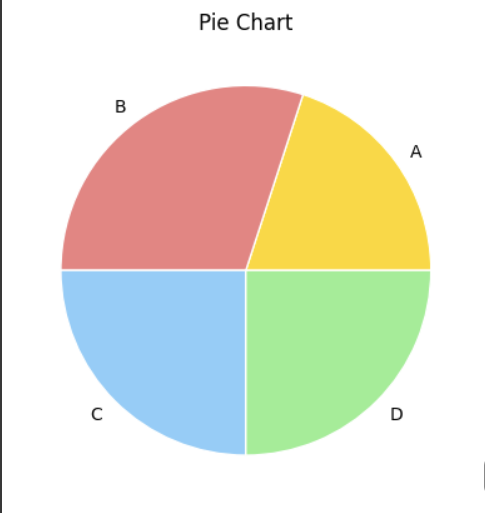
Pie Chart

Use Case: To represent parts of a whole.

Code Snippet:



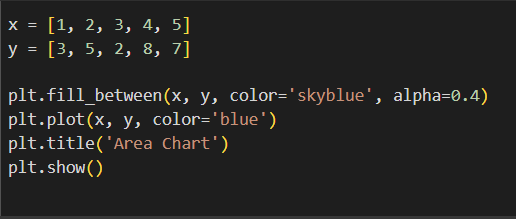
Output:



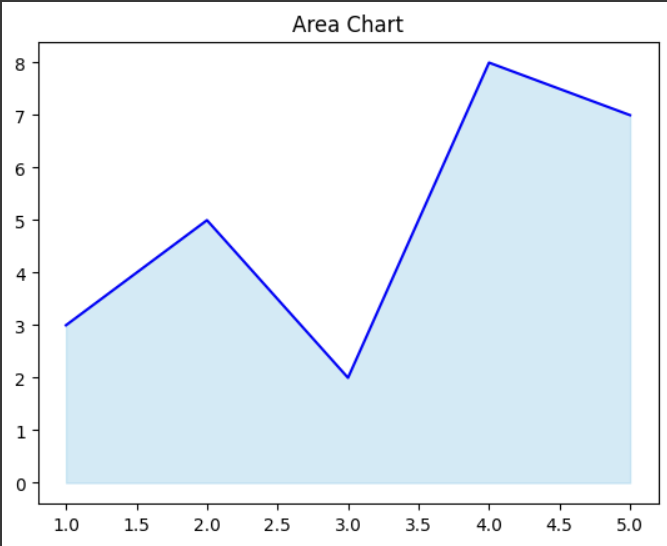
Area Chart

Use case: To visualize quantitative data cumulatively over time or categories.

Code Snippet:



Output:



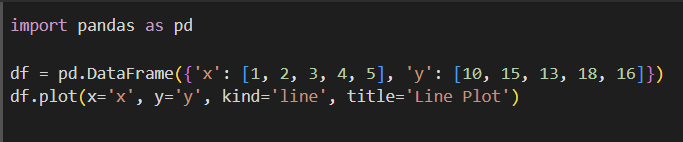
# Pandas Library

Pandas, primarily a data manipulation library, includes built-in plotting capabilities powered by Matplotlib. It offers a quick and convenient way to visualize data directly from DataFrames with df.plot() and its variants.

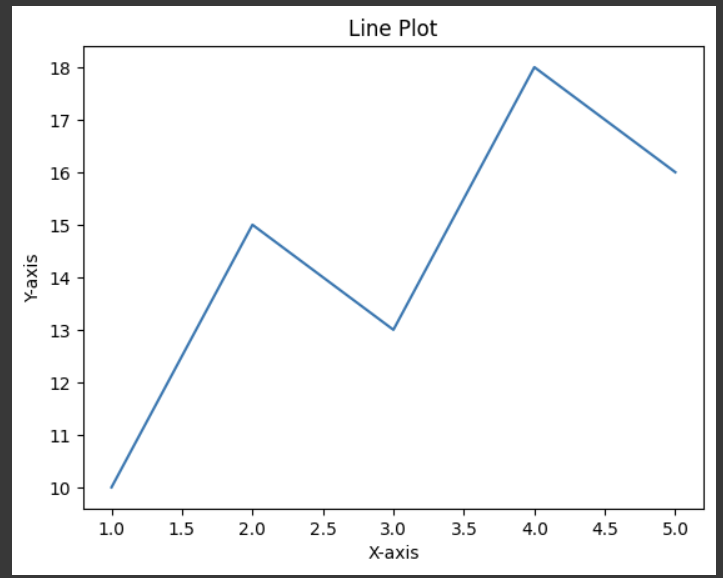
Use Cases of Pandas:

Line Plot

Code Snippet:

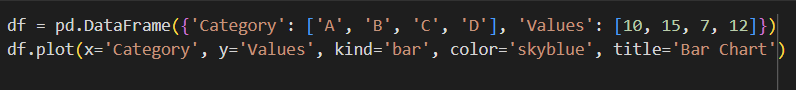


Output:

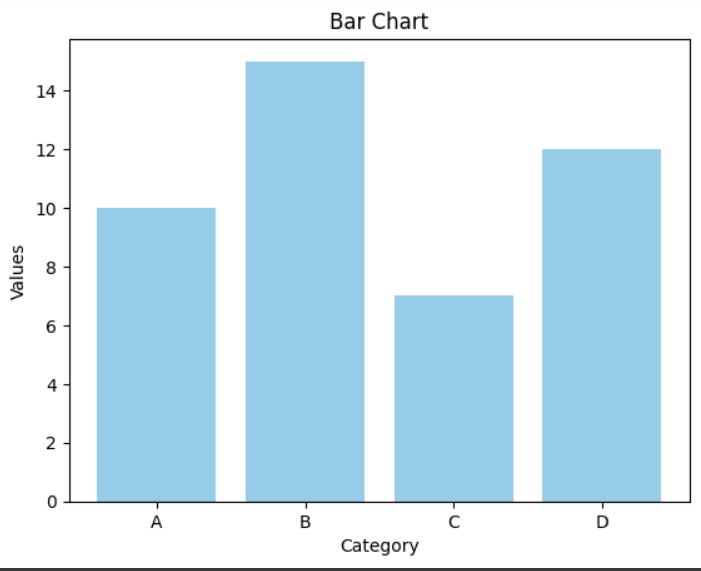


Bar Chart

Code Snippet:

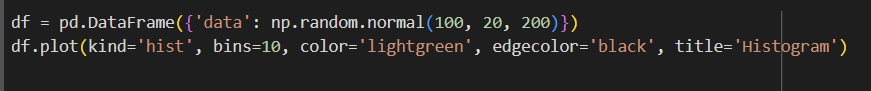


Output:

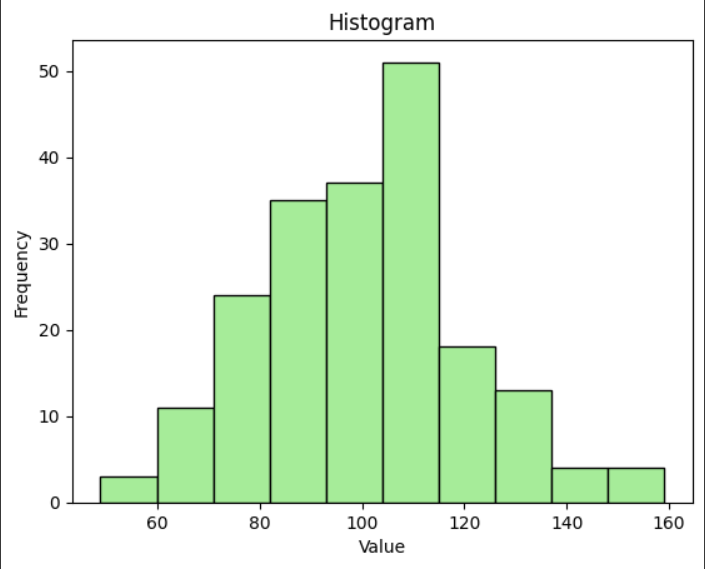


Histogram

Code Snippet:

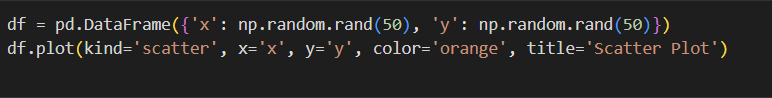


Output:

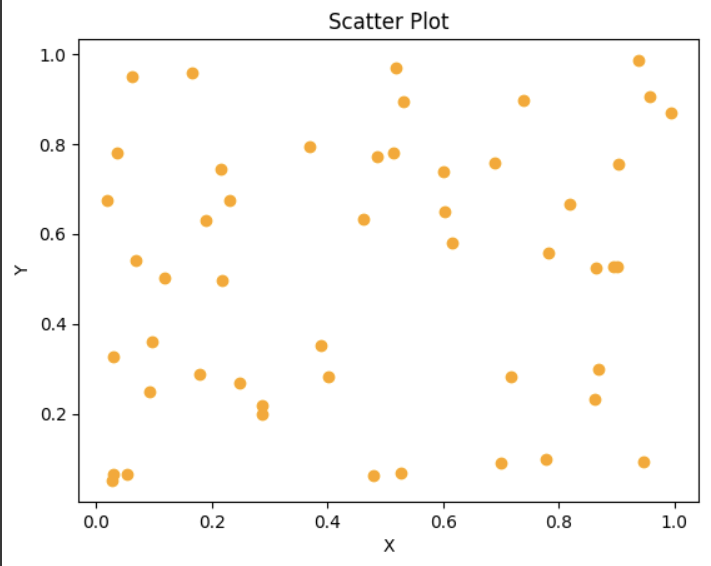


Scatter Plot:

Code Snippet:

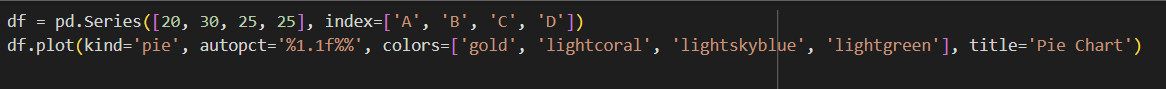


Output:

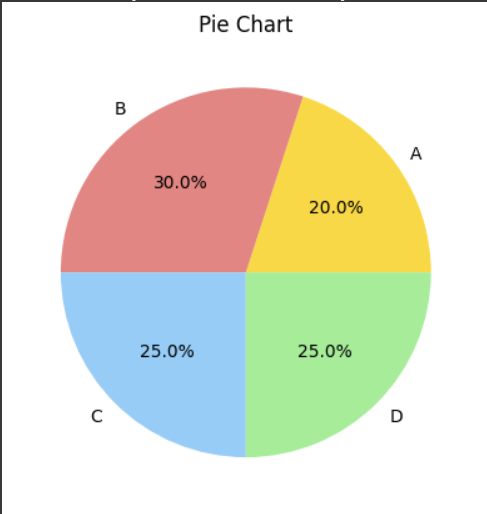


Pie Chart

Code Snippet:

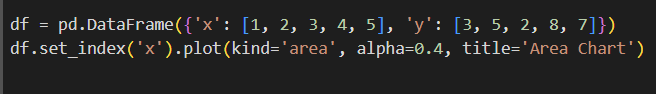


Output:

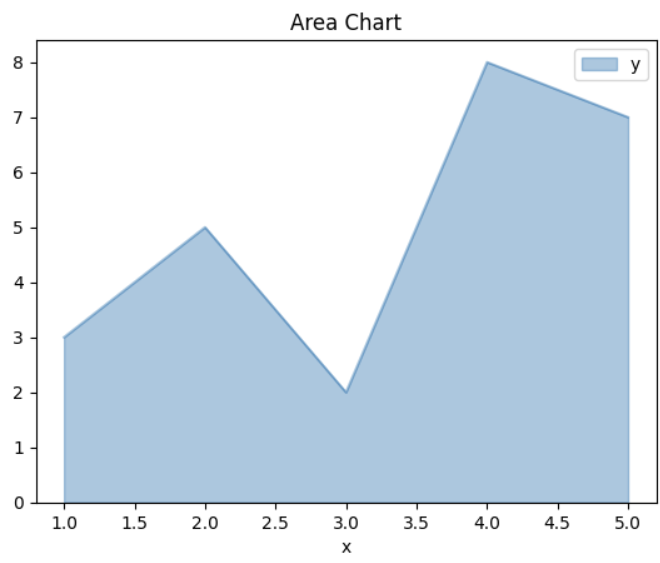


Area Chart:

Code snippet:



Output:



Now that we have seen how the same output has vastly different codes based on the library you are using, we will now compare the two libraries.

# Comparison

|  |  |  |
| --- | --- | --- |
| **Feature** | **Matplotlib** | **Pandas** |
| **Ease of Use** | Requires more setup, flexible | Very simple for quick visualizations |
| **Customization** | Extremely customizable | Limited to standard options |
| **Interactivity** | Static by default; some interactivity possible | Static; limited interactivity |
| **Performance** | Optimized for complex and large plots | Good for small to medium-sized data |
| **Integration** | Works well with NumPy, Pandas, etc. | Built-in with Pandas DataFrames |
| **Best For** | Detailed, complex, customized plots | Fast exploratory and routine visualizations |

# Conclusion:

While Matplotlib offers robust, customizable plotting ideal for detailed reports and publications, Pandas is an excellent tool for quick EDA (Exploratory Data Analysis) directly on your DataFrames.

In real-world scenarios, data scientists often combine both—using Pandas for rapid initial insights and Matplotlib for final polished visuals.